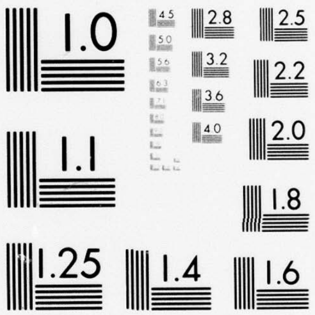
AD-A062 098 GEORGE WASHINGTON UNIV WASHINGTON D C PROGRAM IN LOG--ETC F/G 12/1
USE OF A STABLE GENERALIZED INVERSE ALGORITHM TO EVALUATE NEWTO--ETC(U)
OCT 78 G EMAMI, G P MCCORMICK N00014-75-C-0729
UNCLASSIFIED SERIAL-T-384 NL

1 OF 1
AD
A062098

END
DATE
FILMED
3-79
DDC



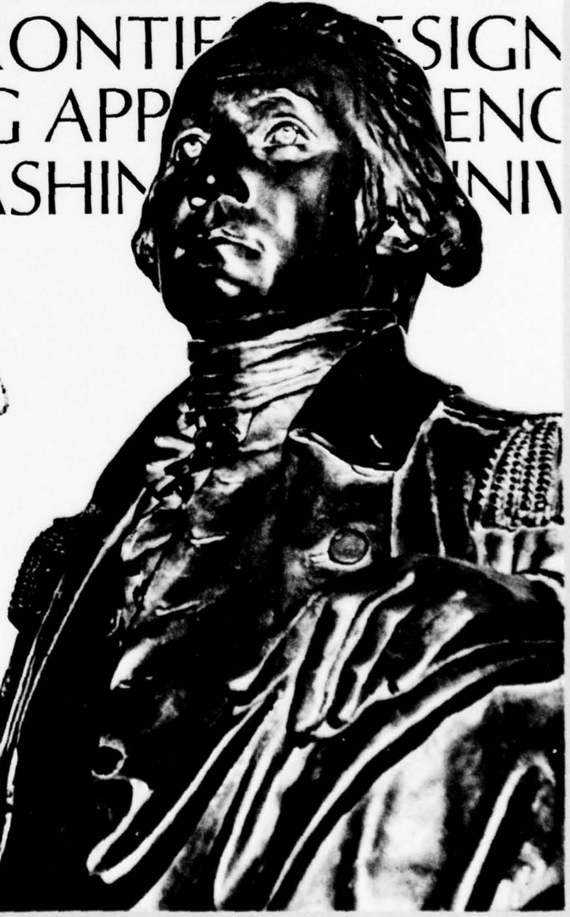
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

12 SC

THE
GEORGE
WASHINGTON
UNIVERSITY

LEVEL II

STUDENTS FACULTY STUDY R
ESEARCH DEVELOPMENT FUT
URE CAREER CREATIVITY CO
MMUNITY LEADERSHIP TECH
NOLOGY FRONTIE ESIGN
ENGINEERING APP ENC
GEORGE WASHIN NIV



INSTITUTE FOR MANAGEMENT
SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING
AND APPLIED SCIENCE

THIS DOCUMENT HAS BEEN APPROVED FOR PUBLIC RELEASE AND SALE; ITS DISTRIBUTION IS UNLIMITED

12

6

USE OF A STABLE GENERALIZED INVERSE ALGORITHM TO EVALUATE NEWTON METHOD STRATEGIES,

by

10

Gholamreza /Emami
Garth P./McCormick

14

Serial T-384

11

25 Oct~~ober~~ 1978

12 62p.

The George Washington University
School of Engineering and Applied Science
Institute for Management Science and Engineering

15 Program in Logistics
Contract N00014-75-C-0729
Project NR 347 020
Office of Naval Research

405 337

NONE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER T-384 ✓	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

USE OF A STABLE GENERALIZED INVERSE ALGORITHM
TO EVALUATE NEWTON METHOD STRATEGIES

5. TYPE OF REPORT & PERIOD COVERED

SCIENTIFIC

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

GHOLAMREZA EMAMI
GARTH P. McMCORMICK

8. CONTRACT OR GRANT NUMBER(s)

N00014-75-C-0729 ✓

9. PERFORMING ORGANIZATION NAME AND ADDRESS

THE GEORGE WASHINGTON UNIVERSITY
PROGRAM IN LOGISTICS ✓
WASHINGTON, D. C. 20037

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

OFFICE OF NAVAL RESEARCH
CODE 434
ARLINGTON, VIRGINIA 22217

12. REPORT DATE

25 OCTOBER 1978

13. NUMBER OF PAGES

57

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)

NONE

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTION OF THIS REPORT IS UNLIMITED.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

UNCONSTRAINED OPTIMIZATION
NEWTON'S METHOD
FACTORABLE FUNCTIONS

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Modifications of Newton's method for minimizing an unconstrained function are presented. A numerically stable matrix factorization is given for obtaining the generalized inverse of an approximation to the positive part of the Hessian matrix and directions of nonpositive curvature. Three different modified Newton strategies are interfaced with the computer program SUMT which transforms a constrained minimization problem → next page

Cont'd

NONE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract continued.

into a sequence of unconstrained problems. Numerical evaluation of these three strategies is given for a variety of test problems.

THE GEORGE WASHINGTON UNIVERSITY
School of Engineering and Applied Science
Institute for Management Science and Engineering

Program in Logistics

Abstract
of
Serial T-384

USE OF A STABLE GENERALIZED INVERSE ALGORITHM TO EVALUATE NEWTON METHOD STRATEGIES

by

Gholamreza Emami
Garth P. McCormick

Modifications of Newton's method for minimizing an unconstrained function are presented. A numerically stable matrix factorization is given for obtaining the generalized inverse of an approximation to the positive part of the Hessian matrix and directions of nonpositive curvature. Three different modified Newton strategies are interfaced with the computer program SUMT which transforms a constrained minimization problem into a sequence of unconstrained problems. Numerical evaluation of these three strategies is given for a variety of test problems.

THE GEORGE WASHINGTON UNIVERSITY
School of Engineering and Applied Science
Institute for Management Science and Engineering

Program in Logistics

USE OF A STABLE GENERALIZED INVERSE ALGORITHM TO EVALUATE NEWTON METHOD STRATEGIES

by

Gholamreza Emami
Garth P. McCormick

1. Introduction

The classical version of Newton's method for minimizing a twice continuously function $f\colon E^n \to E$ takes the form:

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k) \tag{1.1}$$

where $(\nabla^2 f(x_k)$ denotes the n by n Hessian matrix of second derivatives, and $\nabla f(x_k)$ denotes the n by 1 gradient vector of first derivatives evaluated at x_k. Near a point x^* where $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite this algorithm converges with a rate that is at least quadratic (when $\nabla^2 f$ satisfies a Lipschitz condition). When x_0 is started far away from such a point, the algorithm may not converge and if it does, it may not be to a local unconstrained minimizer. Modifications must be made to handle the cases when $\nabla^2 f(x_k)$ is indefinite or singular. It is usually considered desirable to modify the algorithm so that it is a descent method (i.e., $f(x_{k+1}) < f(x_k)$ for all k), and so that accumulation points of the sequence $\{x_k\}$ are stationary points with Hessian matrices which are positive semi-definite.

There are two separate issues. One is to develop strategies for situations where (1.1) does not apply, the other is to provide a numerically stable matrix method to compute the quantities required by the various strategies. In this paper three strategies are presented, and a matrix method is developed for implementing them. The matrix method differs from the traditional ones in that it operates on the Hessian matrix assuming that it is given as the sum of matrices of rank one (outer product matrices) rather than in the square symmetric form that the usual techniques are applied to. This is shown to be a natural byproduct of the factorable programming point of view which is defined in Section 2. The matrix technique is combined with a factorable programming language (McCormick 1974) and the SUMT algorithm (Fiacco and McCormick 1968) for solving general nonlinear programs. Then the three strategies are evaluated on several test problems.

2. Definitions

Most functions of several variables used in nonlinear optimization are complicated compositions of transformed sums and products of functions of a single variable. Once this observation is made formal, an outline of an automatic computer-oriented way of representing nonlinear programming problems becomes clear.

Definition" A function $f(x_1,\ldots,x_n)$ is a *factorable function* of several variables if it can be represented as the last in a finite sequence of functions $\{f_j(x)\}$ which are composed as follows:

$$f_j(x) = x_j \ , \quad \text{for } j=1,\ldots,n \ .$$

For $j > n$, $f_j(x)$ is either of the form

$$f_k(x) + f_\ell(x) \ , \quad k,\ell < j \ ,$$

or of the form

$$f_k(x) \cdot f_\ell(x) \ , \quad k,\ell < j \ ,$$

or of the form

$$T[f_k(x)] \ , \quad k < j \ ,$$

where $T(f)$ is a function of a single variable.

It should be emphasized that this is a natural way of looking at functions of several variables since it corresponds to the way they are used when evaluated at a particular set of values of the variable.

When a function is represented in factorable form, the computation of its first and second derivatives is very simple. Consider the general form for the first and second derivatives of a function which is either the sum of two functions, the product of two functions, or a transformation (function of one variable) of a function. (The derivative is $f'(x)$, a one by n vector, and $f''(x)$ is identical with $\nabla^2 f(x)$) .

For the derivative,

$$[f(x)+g(x)]' = f'(x) + g'(x) \,,$$

$$[f(x)\cdot g(x)]' = f(x) \cdot g'(x) + g(x) \cdot f'(x) \,, \qquad (2.1)$$

and

$$T'[f(x)] = \dot{T}[f(x)]f'(x) \,, \qquad (2.2)$$

where $\dot{T}(f) = \partial T(f)/\partial f$.

For the second derivative,

$$[f(x)+g(x)]'' = f''(x) + g''(x) \,,$$

$$[f(x)\cdot g(x)]'' = f''(x) \cdot g(x) + g''(x) \cdot f(x) \,, \qquad (2.3)$$

$$+ f'(x)^T \cdot g'(x)^T + g(x)^T \cdot f'(x) \,,$$

and

$$[T''[f(x)]] = f''(x) \cdot \dot{T}[f(x)] + f'(x)^T \ddot{T}[f(x)]f'(x) \,, \qquad (2.4)$$

where $\ddot{T}(f) = \partial^2 T(f)/\partial f^2$.

Several points should be noted.

(i) The computation of (2.1) involves $f(x),g(x)$ which would already have been computed in order to evaluate their product.

(ii) The computation of (2.3) involves $f'(x),g'(x)$, which would already have been computed in order to evaluate (2.1).

(iii) Equation (2.4) uses $\dot{T}[f(x)]$ and $f'(x)$, which were previously needed in (2.2).

(iv) By induction, it follows that Hessian matrices of factorable functions are naturally given as sums of *outer products* (*dyads*) of vectors, i.e.,

$$\sum u_i(x)\alpha_i(x)v_i^T(x) + v_i(x)\alpha_i(x)u_i^T(x) \ ,$$

where $\{u_i(x)\},\{v_i(x)\}$ are $n \times 1$ vectors, $\{\alpha_i(x)\}$ are scalars, and the $\{u_i(x)\},\{v_i(x)\}$ are available, having been required for the derivative computation.

Because of the identity

$$ab^T + ba^T = (a+b)(1/2)(a+b)^T + (a-b)(-1/2)(a-b)^T$$

it can be assumed without loss of generality that the Hessian matrix of a factorable function is given in the form

$$F = \nabla^2 f(x) = \sum a_i(x)c_i(x)a_i(x)^T \ .$$

For optimization problems which are factorable, the factorable nonlinear programming problem is written in more compact form:

$$\underset{x \, \varepsilon \, E^n}{\text{minimize}} \ f_N(x)$$

subject to

$$L_i \leq f_i(x) \leq U_i$$

for $i = 1,\ldots,N-1$ (possibly $L_i = -\infty$ and/or $U_i = \infty$) , where $f_i(x) = x_i$, for $i = 1,\ldots,n$, and the remainder are defined recursively as follows: given $\{f_p(x)\}$ for $p = 1,\ldots,i-1$, then for $i = n+1,\ldots,N$,

$$f_i(x) = \sum_{p=1}^{i-1} T_p^i[fp(x)] + \sum_{p=1}^{i-1} \sum_{q=1}^{p} V_{q,p}^i[f_p(x)] \cdot U_{p,q}^i[f_q(x)]$$

where the T's, U's, and V's are functions of a single variable.

A computer program (McCormick 1974) has been written which accepts the factorable problem as coded on cards and provides the interface with nonlinear programming algorithms. A symbolic version of this was used in this study.

A direction s_k is called a *descent direction* at the point x_k if

$$s_k^T \nabla f(x_k) < 0 ,$$

and a *nonascent direction* if

$$s^T \nabla f(x_k) \leq 0 .$$

A direction d_k is called a *direction of negative curvature* if

$$d_k^T \nabla^2 f(x_k) d_k \leq 0 ,$$

and a *direction of nonpositive curvature* if

$$d_k^T \nabla^2 f(x_k) d_k \leq 0 .$$

For a matrix A, the *generalized inverse,* denoted by A^+ is the unique matrix satisfying the four relations:

$$AA^+A = A, \quad A^+AA = A^+ , \quad (AA^+)^T = AA^+ , \text{ and } (A^+A)^T = A^+A .$$

Consider the eigenvector eigenvalue reduction of a symmetric matrix A as

$$A = E \lambda E^T$$

where $EE^T = I$, and λ is a diagonal matrix of eigenvalues. The *positive part* of A denoted by ρ is

$$\rho = \textstyle\sum_{\lambda i>0} e_i \lambda_i e_i^T$$

where e_i is the ith column of E .

The positive part can be shown to be invariant to the particular eigenvector-eigenvalue reduction used.

A symmetric matrix A is called *positive semi-definite singular* (PSDS) if it is positive semi-definite and has at least one zero eigenvalue.

The orthogonal matrix given by

$$G = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

has the effect of rotating the axes through an angle $-\theta$ and is called a Givens matrix. Consider point (X_1,Y_1) in the X-Y plane coordinate of Figure 1.

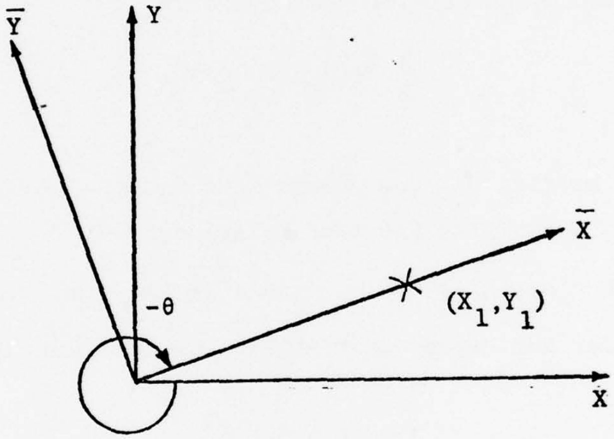


Figure 1

By proper choice of the angle θ, point (X_1,Y_1) can be transformed to $(\bar{X}_1,0)$. A Givens matrix can be used to achieve this type of linear transformation. Consider the linear transformation

$$\begin{pmatrix} \bar{X}_1 \\ \bar{Y}_1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} X_1 \\ Y_1 \end{pmatrix}$$

The desired transformation can be achieved by letting

$$\theta = \cos^{-1} \frac{X_1}{\sqrt{X_1^2 + Y_1^2}}$$

A Givens matrix is usually used to transform the n-vector W

$$W = \begin{pmatrix} W_1 \\ W_2 \\ W_3 \end{pmatrix}$$

where W_1 is a t x 1 vector, W_2 is a scalar and W_3 is (n-t-1) x 1 vector to the general form of $\begin{pmatrix} W_1 \\ \bar{W}_2 \\ \underset{\sim}{0} \end{pmatrix}$ where $\bar{W}_2$ is a scalar.

This transformation can be achieved through a sequence of orthogonal transformations which places a zero successively in the last n-t-1 elements of W. In order to find the reduced form of W we must embed the 2 x 2 Givens matrix in the n x n identity matrix.

Let $G^{i,j}$ denote the matrix which rotates the i,jth axes through an angle $-\theta$, i.e., reduces w_j (jth element of W) to zero by a linear combination of this element with w_i (ith element of W). Let $C = \cos\theta$ and $S = \sin\theta$. Then $G^{i,j}$ is given by

$$G^{i,j} = \left[\begin{array}{ccccccccccccccccc}
1 \\
& \cdot \\
&& \cdot \\
&&& \cdot \\
&&&& 1 \\
&& C &&& 0 &&&&&&&&& S \\
&&&&&& 1 \\
&&&&&&& \cdot \\
&&&&&&&& \cdot \\
&&&&&&&&& \cdot \\
&&&&&&&&&& 1 \\
&& -S &&&&&&&&& 0 &&& C \\
&&&&&&&&&&&& 1 \\
&&&&&&&&&&&&& \cdot \\
&&&&&&&&&&&&&& \cdot \\
&&&&&&&&&&&&&&& \cdot \\
&&&&&&&&&&&&&&&& 1
\end{array}\right]$$

where a blank space represents zero, $C = \frac{W_i}{\sqrt{W_i^2 + W_j^2}}$ and $S = \frac{W_j}{\sqrt{W_i^2 + W_j^2}}$.

Here we show a sequence of Givens matrices which if multiplied by W achieve the derived reduction:

$$\begin{pmatrix} W_1 \\ \bar{W}_2 \\ 0 \end{pmatrix}' = \prod_{i=t+2}^{n} G^{i-1,i} W$$

Consider the vector $W = \begin{pmatrix} 2 \\ 8 \\ 6 \\ 3 \\ 5 \end{pmatrix}$. Through use of Givens matrices,

it is desired to transform W to $\bar{W} = \begin{pmatrix} 2 \\ 8 \\ \bar{W}_2 \\ 0 \\ 0 \end{pmatrix}$, where $\bar{W}_2$ as before is a

scalar to be determined. Then

$$\bar{W}_1 = G^{4,5} W$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{3}{\sqrt{34}} & \frac{5}{\sqrt{34}} \\ 0 & 0 & 0 & \frac{-5}{\sqrt{34}} & \frac{3}{\sqrt{34}} \end{bmatrix} \begin{bmatrix} 2 \\ 8 \\ 6 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \\ 6 \\ \frac{34}{\sqrt{34}} \\ 0 \end{bmatrix}$$

Next,

$$\bar{W} = G^{3,4}\,\bar{W}_1$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{6}{\sqrt{70}} & \frac{17}{\sqrt{595}} & 0 \\ 0 & 0 & \frac{-17}{\sqrt{595}} & \frac{6}{\sqrt{70}} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 8 \\ 6 \\ \frac{34}{\sqrt{34}} \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \\ \frac{70}{\sqrt{70}} \\ 0 \\ 0 \end{bmatrix}$$

In Givens transformation we noted that it required n-t-1 iterations in order to place a zero in the last n-t-1 elements of W. Householder proposed another transformation whereby the entire n-t-1 elements are changed to zero at once. This procedure is much faster than the Givens transformation, and it is preferred whenever possible. It is desired to find a Householder matrix H which can reduce W to $\bar{W} = \begin{pmatrix} W_1 \\ \bar{W}_2 \\ 0 \end{pmatrix}$,

i.e., $\begin{pmatrix} W_1 \\ \bar{W}_2 \\ 0 \end{pmatrix} = H \begin{pmatrix} W_1 \\ W_2 \\ W_3 \end{pmatrix}$.

The Householder matrix is:

$$H = I - \begin{pmatrix} \underset{\sim}{0} \\ \eta \\ W_3 \end{pmatrix} (\varepsilon)^{-1} (\underset{\sim}{0}^T, \eta, W_3^T)$$

where $\eta = W_2 + \sqrt{W_2^2 + ||W_3||^2}$, $\varepsilon = \eta \sqrt{W_2^2 + ||W_3||^2}$ and

$\bar{W}_2 = - \sqrt{W_2^2 + ||W_3||^2}$: As an example consider the Householder transformation required to transform $W = \begin{pmatrix} 2 \\ 8 \\ 6 \\ 3 \\ 5 \end{pmatrix}$ to $\bar{W} = \begin{pmatrix} 2 \\ 8 \\ \bar{W}_2 \\ 0 \\ 0 \end{pmatrix}$. This is given by

$$\eta = 6 + \sqrt{36 + (9+25)} = 6 + \sqrt{70}$$

$$\varepsilon = (6 + \sqrt{70}) \times \sqrt{70} = 6\sqrt{70} + 70$$

$$\bar{W}_2 = - \sqrt{70}$$

$$H = I - \begin{pmatrix} 0 \\ 0 \\ 6+\sqrt{70} \\ 3 \\ 5 \end{pmatrix} (6\sqrt{70} + 70)^{-1} \begin{pmatrix} 0 & 0 & 6+\sqrt{70} & 3 & 5 \end{pmatrix}$$

3. Updating UUT Factorizations

Givens and Householder transformations were originally used to transform a symmetric matrix to a symmetric tridiagonal matrix in order to find its eigenvalues. These orthogonal transformations are now being used in a wide variety of numerical analysis problems. Here, we present a method for modifying a factorization of the form $W = UU^T$ (U is a $t \times t$ upper triangular matrix) to a prespecified form when it is perturbed by a dyad. These updating procedures will be used in Section 4 to introduce a new matrix factorization which is based on the work of Bennett and Green (1966). Here we will consider only three cases which will arise in the process of updating. In Section 4 we present the criteria applicable to each of these cases.

Case 1

Let $W = UU^T$ be perturbed by a dyad (aca^T) that does not change the rank of $\bar{W} = UU^T + aca^T$ from W (where U is a t x t upper triangular matrix, a is a t x 1 vector and c is a scalar which could be either positive or negative). It is desired to factorize $\bar{W} = UU^T + aca^T$ to the form: $\bar{W} = \bar{U}\bar{U}^T$ where $\bar{U}$ is a t x t upper triangular matrix. Consider the expanded form of $\bar{W}$ for $c > 0$:

$$\bar{W} = \left(\begin{array}{cccc|c} U_{11} & U_{12} & \cdots & U_{1t} & a_1\sqrt{c} \\ & U_{22} & \cdots & U_{2t} & a_2\sqrt{c} \\ & & \ddots & \vdots & \\ & & & U_{tt} & a_t\sqrt{c} \end{array}\right)\left(\begin{array}{cccc} U_{11} & & & \\ U_{12} & U_{22} & & \\ \vdots & & \ddots & \\ U_{1t} & U_{2t} & & U_{tt} \\ \hline a_1\sqrt{c} & a_2\sqrt{c} & & a_t\sqrt{c} \end{array}\right)$$

Let the augmented upper triangular matrix be represented by $\hat{U}$, then:

$$\bar{W} = \hat{U}\hat{U}^T$$

Note that by postmultiplying $\hat{U}$ by Givens matrix G, and premultiplying the $\hat{U}^T$ by G^T does not change $\bar{W}$, i.e.,

$$\bar{W} = \hat{U}GG^T\hat{U}^T$$

The proper choice of G is one that when postmultiplied to $\hat{U}$ zeroizes the (t,t+1) element of $\hat{U}$. Repeated application of Givens matrices can eliminate the last column of $\hat{U}$. Also the repeated premultiplying of $\hat{U}^T$ by G^T would result in elimination of the last row of $\hat{U}^T$.

Case 2

Let $W = UU^T$ be perturbed by aca^T such that $\bar{W} = UU^T + aca^T$ has a rank one less than that of W (c is a nonpositive scalar). It is desired to modify $\bar{W} = W + aca^T$ to the form:

$$\bar{W} = G \begin{pmatrix} \bar{U} \\ \hline 0 \end{pmatrix} (\bar{U}^T | 0) G^T$$

where G is an t x t orthogonal matrix and $\bar{U}$ here is a (t-1) x (t-1) upper triangular matrix. The procedure starts as in the previous case, even though the last column of the augmented matrix $\hat{U}$ is a complex vector. The first phase of the procedure is terminated when the last column of augmented matrix $\hat{U}$ is eliminated. During this procedure the first column of $\hat{U}$ also vanishes, i.e., at the conclusion of this phase we have:

$$\bar{W} = \begin{pmatrix} \tilde{U}_{11} & \tilde{U}_{12} & \tilde{U}_{13} & \cdots & \tilde{U}_{1,t-1} \\ \tilde{U}_{21} & \tilde{U}_{22} & \tilde{U}_{23} & & \tilde{U}_{2,t-1} \\ 0 & \tilde{U}_{32} & \tilde{U}_{33} & & \tilde{U}_{3,t-1} \\ \vdots & 0 & & & \vdots \\ 0 & 0 & & & \tilde{U}_{t,t-1} \end{pmatrix} \times \begin{pmatrix} \tilde{U}_{11} & \tilde{U}_{21} & 0 & 0 & 0 \\ \tilde{U}_{12} & \tilde{U}_{22} & \tilde{U}_{32} & 0 & 0 \\ \tilde{U}_{13} & \tilde{U}_{23} & \tilde{U}_{33} & & \\ \vdots & & & & \\ \tilde{U}_{1,t-1} & \tilde{U}_{2,t-1} & \tilde{U}_{3,t-1} & \cdots & \tilde{U}_{t,t-1} \end{pmatrix}$$

The second phase consists of premultiplying the left matrix $\tilde{U}$ with a sequence of Givens matrices such that the element directly below the diagonal vanishes. Also postmultiplying the matrix $\tilde{U}^T$ with a sequence of Givens matrices will yield the desired factorization. As an example of this case, consider the following problem.

Let $W = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 4 & 0 \\ 3 & 5 & 6 \end{pmatrix}$

Find $\bar{W} = W + aca^T$ where $a^T = (3,6,4)$ and $c = -1$. Then $W = \hat{U}_1\hat{U}_1{}^T$.

$$\bar{W} = \begin{pmatrix} 1 & 2 & 3 & 3i \\ 0 & 4 & 5 & 6i \\ 0 & 0 & 6 & 4i \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 4 & 0 \\ 3 & 5 & 6 \\ 3i & 6i & 4i \end{pmatrix}$$

Phase one of this procedure starts by selecting Givens matrices such that the (3,4) position of $\hat{U}_1$ and (4,3) position of $\hat{U}_1{}^T$ vanishes.

$$\bar{W} = \begin{bmatrix} 1 & 2 & 3 & 3i \\ 0 & 4 & 5 & 6i \\ 0 & 0 & 6 & 4i \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{3\sqrt{5}}{5} & \frac{-2\sqrt{5}}{5}i \\ 0 & 0 & \frac{2\sqrt{5}}{5}i & \frac{3\sqrt{5}}{5} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{3\sqrt{5}}{5} & \frac{2\sqrt{5}}{5}i \\ 0 & 0 & \frac{-2\sqrt{5}}{5}i & \frac{3\sqrt{5}}{5}i \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 2 & 4 & 0 \\ 3 & 5 & 6 \\ 3i & 6i & 4i \end{bmatrix}$$

$$\bar{W} = \begin{bmatrix} 1 & 2 & \frac{3\sqrt{5}}{5} & \frac{3\sqrt{5}}{5}i \\ 0 & 4 & \frac{3\sqrt{5}}{5} & \frac{8\sqrt{5}}{5}i \\ 0 & 0 & 2\sqrt{5} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 2 & 4 & 0 \\ \frac{3\sqrt{5}}{5} & \frac{3\sqrt{5}}{5} & 2\sqrt{5} \\ \frac{3\sqrt{5}}{5}i & \frac{8\sqrt{5}}{5}i & 0 \end{bmatrix} = \hat{U}_2\hat{U}_2^T$$

Next we eliminate the (2,4) and (4,2) positions of $\hat{U}_2$ and $\hat{U}_2{}^T$, respectively. It is important to note that in actual computation we need only to consider $\hat{U}$ and find Givens matrix which eliminates selected positions in $\hat{U}$. Then after determining $\bar{U}$, its transpose $\bar{U}^T$ is readily available.

$$\bar{W} = \begin{pmatrix} 1 & 2 & \frac{3\sqrt{5}}{5} & \frac{3\sqrt{5}}{5}i \\ 0 & 4 & \frac{3\sqrt{5}}{5} & \frac{8\sqrt{5}}{5}i \\ 0 & 0 & 2\sqrt{5} & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{5} & 0 & -2i \\ 0 & 0 & 1 & 0 \\ 0 & 2i & 0 & \sqrt{5} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{5} & 0 & 2i \\ 0 & 0 & 1 & 0 \\ 0 & -2i & 0 & \sqrt{5} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 4 & 0 \\ \frac{3\sqrt{5}}{5} & \frac{3\sqrt{5}}{5} & 2\sqrt{5} \\ \frac{3\sqrt{5}}{5}i & \frac{8\sqrt{5}}{5}i & 0 \end{pmatrix}$$

$$\bar{W} = \begin{pmatrix} 1 & \frac{4\sqrt{5}}{5} & \frac{3\sqrt{5}}{5} & -i \\ 0 & \frac{4\sqrt{5}}{5} & \frac{3\sqrt{5}}{5} & 0 \\ 0 & 0 & 2\sqrt{5} & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \frac{4\sqrt{5}}{5} & \frac{4\sqrt{5}}{5} & 0 \\ \frac{3\sqrt{5}}{5} & \frac{3\sqrt{5}}{5} & 2\sqrt{5} \\ -i & 0 & 0 \end{pmatrix} = \hat{U}_3\hat{U}_3^T$$

Continuing the procedure one more time to eliminate the (1,4) position of $\hat{U}_3$, we get

$$\bar{W} = \begin{pmatrix} \frac{4\sqrt{5}}{5} & \frac{3\sqrt{5}}{5} \\ \frac{4\sqrt{5}}{5} & \frac{3\sqrt{5}}{5} \\ 0 & 2\sqrt{5} \end{pmatrix} \begin{pmatrix} \frac{4\sqrt{5}}{5} & \frac{4\sqrt{5}}{5} & 0 \\ \frac{3\sqrt{5}}{5} & \frac{3\sqrt{5}}{5} & 2\sqrt{5} \end{pmatrix} = \hat{U}_4\hat{U}_4^T$$

The second phase of this procedure is used to transform $\bar{W} = \hat{U}_4\hat{U}_4^T$ to $\bar{W} = G^T\tilde{U}\tilde{U}^TG$. In this phase we use Givens matrices which place zero at the (2,1) and (3,2) positions of U_4. Eliminating the (2,1) element of the $\hat{U}_4$ matrix:

$$\bar{W} = \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ \frac{-\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{4\sqrt{5}}{5} & \frac{3\sqrt{5}}{5} \\ \frac{4\sqrt{5}}{5} & \frac{3\sqrt{5}}{5} \\ 0 & 2\sqrt{5} \end{pmatrix} \begin{pmatrix} \frac{4\sqrt{5}}{5} & \frac{4\sqrt{5}}{5} & 0 \\ \frac{3\sqrt{5}}{5} & \frac{3\sqrt{5}}{5} & 2\sqrt{5} \end{pmatrix}$$

$$\times \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ \frac{-\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\bar{W} = \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{4\sqrt{10}}{5} & \frac{3\sqrt{10}}{5} \\ 0 & 0 \\ 0 & 2\sqrt{5} \end{pmatrix} \begin{pmatrix} \frac{4\sqrt{10}}{5} & 0 & 0 \\ \frac{3\sqrt{10}}{5} & 0 & 2\sqrt{5} \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ \frac{-\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\bar{W} = G_1^T \tilde{U} \tilde{U}^T G$$

We notice in this particular example that, while attempting to reduce the (2,1) element of $\hat{U}_4$ to zero, the (2,2) element also vanishes. Therefore in order to transform $\bar{W}$ to the required factorization, we should use a permutation matrix which exchanges the second and third rows of $\bar{U}_5$ and hence yields

$$\bar{W} = \begin{pmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{-\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{4\sqrt{10}}{5} & \frac{3\sqrt{10}}{5} \\ 0 & 2\sqrt{5} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{4\sqrt{10}}{5} & 0 & 0 \\ \frac{3\sqrt{10}}{5} & 2\sqrt{5} & 0 \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \\ \frac{-\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{pmatrix}$$

Case 3

Here we modify $W = \begin{pmatrix} U \\ 0 \end{pmatrix}(U^T, 0)$ by a dyad bcb^T where b is a $n \times 1$ vector, c is a scalar and $t < n$. It is desired to transform $\bar{W} = \begin{pmatrix} U \\ 0 \end{pmatrix}(U^T, 0) + bcb^T$ to the form of

$$\bar{W} = H^T \left(\frac{\bar{U}}{0}\right) (\bar{U}^T | 0) H$$

where H is an $n \times n$ orthogonal matrix and $\bar{U}$ is an upper triangular matrix of rank $t + 1$.

$$\begin{pmatrix} U_{11} & U_{12} & \cdots & U_{1t} & b_1\sqrt{c} \\ & U_{22} & \cdots & U_{2t} & b_2\sqrt{c} \\ & & & \vdots & \vdots \\ & & & U_{tt} & b_t\sqrt{c} \\ & & & & b_{t+1}\sqrt{c} \\ & & & & \vdots \\ & & & & b_n\sqrt{c} \end{pmatrix} \begin{pmatrix} U_{11} & & & & & & \\ U_{12} & U_{22} & & & & & \\ \vdots & \vdots & & & & & \\ U_{1t} & U_{2t} & \cdots & U_{tt} & & & \\ b_1\sqrt{c} & b_2\sqrt{c} & \cdots & b_t\sqrt{c} & b_{t+1}\sqrt{c} & \cdots & b_n\sqrt{c} \end{pmatrix}$$

Let $\bar{W} = \bar{Z}\bar{Z}^T$ and let H be a Householder matrix which can reduce the (t+2)th through the nth position of the last column of $\bar{Z}$ to zero, i.e.,

$$H\bar{Z} = \left(\frac{\bar{U}}{0}\right) .$$

Due to the symmetry of H, $\bar{Z}^T H = (\bar{U}^T | 0)$. Then the desired factorization is given:

$$\begin{aligned} \bar{W} &= H^T H \bar{Z}\bar{Z}^T H^T H \\ &= H^T \left(\frac{\bar{U}}{0}\right) (\bar{U}^T | 0) H \end{aligned}$$

4. Estimating the Positive Part of the Hessian Matrix and Directions of Nonpositive Curvature

In this section we will give a general algorithm for estimating the positive part of the Hessian matrix and as a byproduct direction of nonpositive curvature. The Hessian matrix is assumed in outer product or dyadic form, i.e.

$$\nabla^2 f(x_k) = F^{(k)} = \sum_{j=1}^{s} a_j^k c_j^k \left(a_j^k\right)^T .$$

First a few preliminary results are required. In the following, the matrix A will be assumed positive semi-definite, a is an n by 1 vector, e is n by 1, and c is a scalar.

If $Ae = 0$, then

$$[A+Az\alpha z^T A]e = 0 \tag{4.1}$$

for all vectors z , all scalars α .

If $a^T A^+ \neq 0$, and $[I - AA^+]a = 0$, then

$$[A+a(-a^T A^+ a)^{-1} a^T]A^+ a = 0 . \tag{4.2}$$

A necessary and sufficient condition that $[I-AA^+]a = 0$ is that $a^T e = 0$ for all vectors e where $Ae = 0$. (4.3)

A necessary condition for $A + aca^T$ to be positive semi-definite is that

$$1 + ca^T A^+ a \geq 0 . \tag{4.4}$$

If $[I-AA^+]a = 0$, then (4.4) is a sufficient condition for $A + aca^T$ to be positive semi-definite. (4.5)

For simplicity of notation, the superscript (k) is omitted from the general description of the algorithm. Let P denote the estimate of the positive part of F , and P_j^+ denote its generalized inverse of the beginning of iteration j . Assume the dyads are relabelled so that

$R = \{1,\ldots,r\}$ and $S = \{r+1,\ldots,s\}$, with $c_j > 0$, $j \varepsilon R$, $c_j < 0$, $j \varepsilon S$. Then

$$F = \sum_{j=1}^{r} a_j c_j a_j^T + \sum_{j=r+1}^{s} a_j c_j a_j^T \ .$$

The dyads with positive scalars are treated first, and then as many of the dyads with negative scalars are "absorbed" until further absorption is impossible.

```
Set $P_1 = 0$ , $j = 0$ .
    A. Set $j = j + 1$ . If $j > r$ go to B .
       Set $P_{j+1} = P_j + a_j c_j a_j^T$ .                                    (4.6)
       Compute $P_{j+1}^+$ (see Case(2(b)) ) .
       Return to A .
    B. If $j > s$ , go to C .
       If $[I-P_j P_j^+]a_j \neq 0$ , set $P_{j+1} = P_j$ . Set $j = j+1$ ,
          go to B .
       If $[I-P_j P_j^+]a_j = 0$ , calculate $K_j = 1 + c_j a_j^T P_j^+ a_j$ .
       If $K_j \geq 0$ , set $P_{j+1} = P_j + a_j c_j a_j^T$ . Set $S = S - \{j\}$ .
       Compute $P_{j+1}^+$ (see (case(1a) or a(b)) . Set $j = j + 1$ ,
          go to B .                                                              (4.7)
       If $K_j < 0$ , set $j = j + 1$ , go to B . (For a possible
          partial absorption of the dyad see the discussion in
          Section 8).
    C. Now $P_{s+1}$ is an estimate of the positive part of F . It
       is a positive semi-definite matrix. If it is not positive
       definite, directions of nonpositive or negative curvature
```

are of the form $\left[I-P_{s+1}\, P^{+}_{s+1}\right]z$ for any z such that there exists an $i(z) \in S$ where $a^{T}_{i(z)}\left[I-P_{s+1}P^{+}_{s+1}\right] z \neq 0$.

To see this,

$$z^{T}\left[I-P_{s+1}P^{+}_{s+1}\right] F \left[I-P_{s+1}P^{+}_{s+1}\right]z$$

$$= z^{T}\left[I-P_{s+1}P^{+}_{s+1}\right] \left[P_{s+1}+ \textstyle\sum_{i\varepsilon s} a_{i}c_{i}a^{T}_{i}\right] \left[I-P^{+}_{s+1}P_{s+1}\right]z \qquad (4.8)$$

$$\leq \left\{z^{T}\left[I-P_{s+1}P^{+}_{s+1}\right]a_{i(z)}\right]^{2} c_{i(z)} < 0 .$$

If $j \varepsilon S$ at the end of the iterations it needs to be shown that

$$\left[I-P_{s+1}P^{+}_{s+1}\right]a_{j} \neq 0 \qquad (4.9)$$

(which then implies that $P^{+}_{s+1}\, a_{j}$ is a direction of negative curvature).

There are two possibilities for j being in S . First, $\left[I-P_{j}P^{+}_{j}\right]a_{j} \neq 0$. By (4.3) there is an $\bar{e}$ such that $P_{j}\bar{e} = 0$ and $a^{T}_{j}\bar{e} \neq 0$. The only updating to P_{k} from iteration to iteration is when $\left[I-P_{k}P^{+}_{k}\right]a_{k} = 0$ which implies that if $P_{k+1} \neq P_{k}$, $P_{k+1} = P_{k} + P_{k}z\, \gamma\, z^{T}P_{k}$. From (4.1) it follows by induction that $P_{s+1}\bar{e} = 0$. Thus $\left[I-P_{s+1}P^{+}_{s+1}\right]a_{j} \neq 0$.

The second case is when $\left[I-P_{j}P^{+}_{j}\right]a_{j} = 0$ and $K_{j} < 0$. Now by (4.2), $P_{j+1}P^{+}_{j}a_{j} = 0$. Also, $a^{T}_{j}\, P^{+}_{j}a_{j} \neq 0$. Therefore by (4.3),

$$\left[I-P_{j+1}P^{+}_{j+1}\right]a_{j} \neq 0 .$$

The same argument as above yields (4.9).

In the specific realization of this algorithm given next, directions of nonpositive curvature and negative curvature will be more clearly defined.

Now we present a numerically stable method for computing the generalized inverse of approximate positive part of the Hessian matrix. This method does the linear algebra required by steps (4.6) and (4.7) of the general algorithm just presented. At each iteration j (which is suppressed for notational simplicity) P is assumed to be of the form

$$P = Q^T \left(\frac{U}{0}\right) (U^T|0)Q$$

where Q is an n by n orthogonal matrix, and U is a t x t upper triangular matrix of full rank. The generalized inverse of P is therefore

$$P^+ = Q^T \left(\frac{U^{-T}}{0}\right) (U^{-1}|0)Q \ .$$

There is never any reason to compute this explicitly.

Initially P_0 has a Q equal to the identity matrix, and U has rank zero. The following cases for the updating procedure are in slightly different order than presented in the preceding general algorithm. The appropriate correspondences are indicated.

<u>Algorithm</u>

Assume $P = Q^T \left(\frac{U}{0}\right) (U^T|0)Q$ is known where Q is an n x n orthogonal matrix and U is t x t full rank upper triangular matrix. It is desired to find whether $\overline{P} = P + aca^T$ is positive semidefinite for some n x 1 vector a and scalar c ; and when $\overline{P}$ is positive semidefinite to compute the QTU factorization of $\overline{P}$.

<u>QTU Algorithm</u>

Compute the vector $\left(\frac{\omega}{\beta}\right)$

$$\left(\frac{\omega}{\beta}\right) = Qa\sqrt{c}$$

where ω is t x 1 and β is (n-t) x 1 are real or complex vectors depending on whether c is positive or negative. There are two cases to be considered.

Case 1

The vector $\beta = \ell$. This implies $(I-PP^{+})a = 0$ since

$$Q(I-PP^{+})a\sqrt{c} = Qa\sqrt{c} = QQ^{T}\left(\frac{U}{0}\right)(U^{T}|0)QQ^{T}\left(\frac{U^{-T}}{0}\right)(U^{-1}|0)Qa\sqrt{c}$$

$$= Qa\sqrt{c} - \left(\begin{array}{c|c} I_{txt} & 0 \\ \hline 0 & 0 \end{array}\right) Qa\sqrt{c} \tag{4.10}$$

$$= \left(\frac{\omega}{\beta}\right) - \left(\frac{\omega}{0}\right).$$

It can be seen that when $\beta = 0$, (4.10) is equal to zero. Note that this case also embodies the situation when U is an $n \times n$ upper triangular matrix, i.e., $t = n$.

Next, compute $K = 1 + ca^{T}P^{+}a$. There are three possibilities which need to be considered.

a) If $K > 0$, then from (4.5), it follows that $\overline{P} = P + aca^{T}$ is positive semidefinite, and the rank of $\overline{P}$ is equal to t . (This corresponds to (4.7) when $K_j > 0$) .

$$\overline{P} = Q^{T}\left(\frac{U}{\ell}\right)(U^{T}|\ell)Q + aca^{T}$$

$$= Q^{T}\left(\frac{U}{\ell}\right)(U^{T}|\ell)Q + Q^{T}\left(\frac{\omega}{\ell}\right)(\omega^{T}|\ell)\,Q \tag{4.11}$$

$$= Q^{T}\left[\left(\frac{U}{\ell}\right)(U^{T}|\ell) + \left(\frac{\omega}{\ell}\right)(\omega^{T}|\ell)\right]Q \quad .$$

Then using Givens transformation (see Section 2) we can transform

(4.11) to

$$\overline{P} = Q^{T}\left(\frac{\overline{U}}{\ell}\right)(\overline{U}^{T}|\ell)Q$$

(where $\overline{U}$ is a $t \times t$ upper triangular matrix).

b) If $K = 0$, from (4.1), (4.2) and (4.5) it follows that $\bar{P} = P + aca^T$ is positive semidefinite and the rank of $\bar{p}$ is $t - 1$. The rank reduction occurs because a positive eigenvalue now becomes zero. Givens transformations could be used to convert (4.11) to

$$\bar{P} = Q^T \left[\begin{array}{c|c} G^T & 0 \\ \hline \ell & I \end{array}\right] \left(\frac{\bar{U}}{\ell}\right) (\bar{U}^T | 0) \left[\begin{array}{c|c} G & 0 \\ \hline \ell & I \end{array}\right] Q$$

where G is a $t \times t$ orthogonal matrix and $\bar{U}$ is a $(t-1) \times (t-1)$ upper triangular matrix.

Let $\bar{Q}^T = Q^T \left[\begin{array}{c|c} G^T & 0 \\ \hline 0 & I \end{array}\right]$, i.e., $\bar{P} = \bar{Q}^T \left(\frac{\bar{U}}{0}\right) (\bar{U}^T | 0)\bar{Q}$.

(This corresponds to (4.9) when $k_j = 0$.)

c) When $K < 0$ the matrix $\bar{P}$ is not positive semidefinite. No modification is made although as discussed in Section 8, part of the dyad could be absorbed.

Case 2

In this case vector β is not a zero vector, (i.e. $[I-PP^+]a \neq 0$.

a. If $c < 0$ then $\bar{P}$ is not positive semidefinite and, consequently, there is no need to update. (This corresponds to the first line following β .)

b. If $c > 0$, $\bar{P}$ is positive semidefinite; then

$$\begin{aligned} \bar{P} &= Q^T \left(\frac{U}{\ell}\right) (U^T | \ell) Q + aca^T \\ &= Q^T \left(\frac{U}{\ell}\right) (U^T | \ell) Q + Q^T \left(\frac{\omega}{\beta}\right) (\omega^T | \beta^T) Q \qquad (4.12) \\ &= Q^T \left[\begin{array}{c|c} U & \omega \\ \hline \ell & \beta \end{array}\right] \left[\begin{array}{c|c} U^T & \ell \\ \hline \omega^T & \beta^T \end{array}\right] Q \ . \end{aligned}$$

To convert (4.12) to the proper form a Householder transformation can be used to make the following transformation.

$$H\left[\begin{array}{c|c} \bar{U} & \omega \\ \hline 0 & \beta \end{array}\right] = \left[\begin{array}{c|c} U & \omega \\ \hline 0 & r \\ \ell & \ell \end{array}\right]$$

where r is a scalar.

Letting $\bar{Q} = HQ$ and $\bar{U} = \left[\begin{array}{c|c} U & \omega \\ \hline 0 & r \end{array}\right]$, then

$\bar{P} = \bar{Q}^T \left(\frac{\bar{U}}{\ell}\right) (\bar{U}^T|\ell)\bar{Q}$ where $\bar{U}$ is a (t+1) by (t+1) upper triangular matrix. It is apparent that whenever $\beta = \left(\frac{r}{\ell}\right)$, where r is scalar, then H is equal to the identity matrix. (This corresponds to (4.6)).

After the Hessian matrix has been processed according to the two previously defined algorithms the situation is

$$F = Q^T \left(\frac{U}{0}\right) (U^T|0)Q + \textstyle\sum_{j\varepsilon S} a_j c_j a_j^T \ .$$

The matrix to the right of the equals sign is an estimate of the positive part of F . If $S = \emptyset$, and if U is n by n , then F is positive definite. If $S = \emptyset$ and U is t by t with $t < n$, F is positive semidefinite singular and any of the last (n-t) rows of Q are directions of zero curvature.

When $S \neq \emptyset$, any of the last (n-t) rows of Q is a direction of nonpositive curvature, and at least one of them is a direction of negative curvature. To see this, note that

$$[I-P_{s+1}P_{s+1}^{+}]a_j = Q_2^T Q_2 a_j$$

where Q_2 is the matrix consisting of the last (n-t) rows of Q , and a_j is any vector with $j\varepsilon S$. From (4.9), it follows that a_j^T is not orthogonal to at least one of the rows of Q_2. This row is therefore a direction of negative curvature.

5. Modified Newton Strategies

In this section we outline three modified Newton strategies. We use the QTU factorization for eavaluating the approximation to the positive part of the Hessian matrix and directions of nonpositive curvature. In Section 7 these strategies are evaluated.

Let x_0 be a given initial point; then at the kth iteration x_k is known and x_{k+1} needs to be determined.

Strategy 1 (S1MNM)

If the Hessian matrix at x_k is positive definite, set $s_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$, and find x_{k+1} using the optimal step size procedure (OSSP) (see (5.1)). If it is not positive definite, select d_k, a nonascent direction of nonpositive curvature. Set $s_k = d_k$ and find x_{k+1} using OSSP. There are many different ways d_k can be chosen and these are discussed later in this section.

The motivation for this algorithm is that when $\nabla^2 f(x_k)$ is not positive definite moving along a direction of nonpositive curvature will tend to move the sequence of points into a region where the Hessian is positive definite so that the usual Newton move will apply. Even when the Hessian matrix is positive definite, it is deemed advantageous to use the optimal step size procedure instead of using a step of size of one as prescribed by the classical method.

Strategy 2 (S2MNM)

This is the same as S1MNM except that following the move along d_k in the nonpositive definite case another move is made in the direction $-P_k^+ \nabla f(x_k)$. Formally, let $y_k = x_k + d_k t_k$ where t_k solves OSSP.

Then find τ_k solving $\min_{\tau \geq 0} f[y_k - P_k^+ \nabla f(x_k)\tau]$. Set $x_{k+1} = y_k - P_k^+ \nabla f(x_k)\tau_k$. This extra step in the iteration is an attempt to minimize some portion of the function even though it doesn't act like a strictly convex function based on the information at x_k .

Strategy 3 (S3MNM)

Let $0 < \alpha < 1$ be a given constant. If the Hessian matrix at x_k is positive definite, set $s_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$, and $d_k = 0$. Find x_{k+1} using the Second Order Armijo Step Size procedure given below. If the Hessian is not positive definite, set $s_k = -P_k^+ \nabla f(x_k)$, select d_k , a direction of nonpositive curvature, and use the Second Order Armijo Step Size procedure (5.2) to find x_{k+1} . This can be thought of as a way of combining a move which attempts directly to minimize f with a move which attempts to get the sequence in a region where the Hessian matrix is positive definite.

The Optimal Step Size Procedure (OSSP)

Given s_k , a nonascent direction, set

$$x_{k+1} = x_k + s_k t_k \tag{5.1}$$

where t_k solves

$$\min_{t \geq 0} f[x_k + s_k t] .$$

Second Order Armijo Step Size Procedure

Let α be a given constant where $0 < \alpha < 1$. Assume at x_k are given s_k , a nonascent direction, and d_k a nonascent direction of nonpositive curvature. Define

$$y_k(i) = x_k + s_k 2^{-i} + d_k 2^{-i/2} .$$

Find i(k), the smallest integer from $i = 0,1,\ldots$ such that

$$f[y_k(i)] - f(x_k) \leq \alpha 2^{-i}[s_k^T \nabla f(x_k) + \frac{1}{2} d_k^T \nabla^2 f(x_k) d_k] . \tag{5.2}$$

Set $x_{k+1} = y_k[i(k)]$.

The above three strategies all require a direction of nonpositive curvature. The QTU algorithm produces some as the last (n-t) rows of Q . For any $i \varepsilon S$ (after all possible dyads have been absorbed), any $\underline{+}P_{s+1}a_i$ also a direction of negative curvature. Several strategies for selecting from these were evaluated. Let $d^1,\ldots,d^1$ be ℓ available nonascent directions of nonpositive curvature.

DNPC1: Set $d = \sum_{i=1}^{\ell} d^i/\ell$ (the average of the possibilities).

DNPC2: Set $d = \underline{+}P_{s+1}a_I$ where a_I minimizes $1 + c_i a_i^T P_{s+1}^+ a_i$ for $i \varepsilon S$.

If S is empty, DNPC3 is used.

DNPC3: Set $d = d^I$ where d^I minimizes $(d^i)^T F d^i$ for $i=1,\ldots,\ell$.

DNPC4: Set $d = d^I$ where d^I minimizes $(d^i)^T \nabla f(x_k)$ for $i=1,\ldots,\ell$.

6. Statement of Problems Solved

In this section is given the statement of the problems solved by the modified Newton methods. In Section 7 is given the work required to solve them using the different options. Some of the problems are constrained and the SUMT method for solving constrained problems using a sequence of unconstrained problems (to which the methodology applies) is briefly presented.

Consider the general nonlinear programming problem in its canonical form. Find $X \varepsilon R^n$ by solving

$$\text{minimize } f(X)$$

subject to

$$g_i(X) \geq 0 \quad i=1,\ldots,m$$

$$h_j(X) = 0 \quad j=1,\ldots,P .$$

Among the several existing classes of techniques for solving the general nonlinear programming problem are methods which transform a constrained problem into a sequence of unconstrained problems. Here we only briefly discuss the method of Fiacco and McCormick (1968) which was used in this paper to solve a general nonlinear programming problem. The proposed transformed function which is known as the penalty function is defined as:

$$P(X,r) = f(X)-r \sum_{i=1} \ell n\ g_i\ (X) + \sum_{j=1}^{p} h_1^2\ (X)/r$$

If $\{r_k\}$ is an infinite sequence of points such that $\lim_{k\to\infty} r_k = 0$ then, loosely stated, there exists a point for each k that minimizes $p(X,r_k)$, $X^{(k)}$; every limit point of $X^{(k)}$ is an optimal solution of the original constrained problem. The reader is referred to Fiacco and McCormick for a comprehensive statement and proof of the convergence theorems.

Here, following Fiacco and McCormick we summarize the general iteration of this algorithm. Let $R^0 \equiv \{X|;\ g(X)>0,\ i=1,\ldots,m\}$.

i. Find a point X^0 in R^0 .

ii. Select the initial value of r , r_1 . The selection of r_1 may be arbitrary or based on any one of the criteria described in Section 7.1 of Fiacco and McCormick (1968).

iii. Find an unconstrained minimum of $P(X,r)$ for the current value of r_k . The method used in this paper to solve this unconstrained problem is the Modified Newton method described in Section 5.

iv. Estimate the solution of the original constrained problem by extrapolation.

v. Stop if the current solution satisfies the convergence criteria, otherwise select $r_{k+1} < r_k$ and continue the procedure from step iii.

We now present the solution of several test problems which were solved by the Modified Newton method described in Section 5. The purpose of presenting these problems and their solutions in this section is to establish the credibility of the QTU algorithm since the solutions to these problems were obtained by other methods also. As we discussed previously, the solution procedure for general nonlinear programming problems which is used in this paper is based on the solution of a sequence of unconstrained functions. The computer program, which was coded and used here, requires the user to transform the unconstrained problem by explicitly defining the penalty function (P(X,r)) as the objective function.

Shell Dual

This problem was formulated by the Shell Development Company for the purpose of testing nonlinear programming routines.

Minimize:
(X,Y)

$$f \equiv 2\sum_{i=1}^{5} d_i X_i^3 + \sum_{k=1}^{5} \left(\sum_{j=i}^{5} F_{kj} X_j \right)^2 - \sum_{j=1}^{10} b_j Y_j$$

subject to:

$$g_i \equiv e_i + 2\sum_{k=i}^{5} F_{ki} \left(\sum_{j=1}^{5} F_{kj} X_j \right) + 3d_i X_i^2 - \sum_{j=i}^{10} a_{ji} Y_j > o \quad \text{for } i=1,\dots,5$$

$$g_{i+5} \equiv X_i \geq o \quad \text{for } i=1,\dots,5$$

$$g_{j+10} \equiv Y_j \geq o \quad j=1,\dots,10$$

The values of a_{ij}, b_i, d_i, e_i and F_{ij} are given in Table 2. The solution to this problem is:

X_1 = .3000021, X_2 = .3334455, X_3 = .4000091

X_4 = .4283571, X_5 = .2240632, X_6 = .1683284E-05

X_7 = .3128900E-04, X_8 = .5174274E 01, X_9 = .4377716E-04

X_{10} = .3061098E 01, Y_1 = .1184054E 02, Y_2 = .1594590E-05

Y_3 = .1076526E-05, Y_4 = .1043444, Y_5 = .8907809E-04

TABLE 2

DATA FOR SHELL DUAL

j i

a_{ij}					b_i
-16	2	0	1	0	-40
0	-2	0	0.4	2	-2
-3.5	0	2	0	0	-.25
0	-2	0	-4	-1	-4
0	-9	-2	1	-2.8	-4
2	0	-4	0	0	-1.
-1	-1	-1	-1	-1	-40
-1	-2	-3	-2	-1	-60
1	2	3	4	5	5
1	1	1	1	1	1

j i

F_{ij}				
0	1	0	-1	0
3	2	-3	2	3
-2	3	0	-3	2
-4	5	0	-5	4
1	0	-1	0	1

d_j	4	8	10	6	2
e_j	-15	-27	-36	-18	-12

Gasoline Demand Function

A nonlinear dynamic demand function was formulated and its parameters were estimated using the technique of least squares. Let $q_{i,j}$ = the demand for gasoline per capita in ith census region in jth year.[1]

$P_{i,j}$ = average price in census region i for year j[2]

$D_{i,j}$ = disposable personal income per capita for region in year j[3]

M_j = average miles per gallon consumed per passenger car[4]

For this study we considered 3 census regions over a 9 year period, i.e. i=1,2,3, and j=1,2,...,9. The demand function considered was:

$$q_{i,j} = q_{i,j-1} \exp[-X_1 P_{i,j} + X_2 P_{i,j-1} + X_3 D_{i,j} - X_4 D_{i,j-1} + X_5] -X_6 M_j^{X_7} - X_{i+7}$$

Table 3 provides the data which was used in this model.

The nonlinear least squares problem which was solved is:

$$f \equiv \sum_{i=1}^{3} \sum_{j=2}^{9} \{q_{i,j} - q_{i,j-1} \exp [-x_1 P_{ij} - X_2 P_{i,j-1} + X_3 D_{i,j} - X_4 D_{i,j-1} + X_5] + X_6 M_j^{X_7} - X_{i+7}\}^2$$

The solution that was obtained for this model is:

X_1 = .6668957E-02, X_2 = .5533945E-02, X_3 = .1442763E-01

X_4 = .1707499E-01, X_5 = .8533931E-01, X_6 = .1050872E 04

X_7 = -.2894233E-03, X_8 = .2003835E 02, X_9 = .2350300E 02

X_{10} = .2460730E 02

[1] Source: American Petroleum Institute

[2] Source: Platt's OILGRAM price service

[3] Source: U.S. Department of Commerce, Bureau of Economic Analysis

[4] Source: U.S. Highway Statistics

TABLE 3

DATA FOR GASOLINE MODEL

region	(Consumption/Capita) in Gallons			(Disposable Consumption/ Capita x 100) in Dollars		
year	1	2	3	1	2	3
1967	391.98	342.33	382.45	20.65	29.83	24.34
1968	412.39	361.13	403.84	21.4656	30.60	25.24
1969	437.68	374.31	427.27	21.9004	30.86	25.83
1970	455.86	395.28	447.19	22.3686	31.50	26.56
1971	480.13	406.07	468.80	23.2121	31.79	27.46
1972	511.38	424.49	479.68	24.4494	32.36	28.86
1973	533.24	433.06	523.73	25.8538	33.44	30.41
1974	516.65	423.21	504.57	25.3773	32.85	29.69
1975	527.57	428.98	510.36	24.8000	32.60	28.90

	Price, Cents/Gallons; In Constant Dollar Adjusted by CPI (1967 = 100)			Average Miles/ Gallon	Consumer Price Index
1967	33.08	33.07	32.68	14.05	100
1968	32.46	33.00	32.71	13.91	104.2
1969	31.61	32.77	32.50	13.75	109.8
1970	30.72	31.77	31.58	13.7	116.3
1971	30.12	31.39	31.07	13.73	121.3
1972	28.83	29.94	30.28	13.67	125.3
1973	29.08	30.09	30.11	13.29	133.1
1974	35.37	36.70	37.44	13.71	147.1
1975	34.53	35.27	35.43	13.81	161.2

Multi-item Continuous Review Inventory Model

Schrady and Choe (1971) formulated a multi-item, continuous review inventory model with back orders. The formulation involved the minimization of total time-weighted units short per unit time when the lead time demand for the ith item was normally distributed with mean μ_i and variance σ_i^2. The constraints which were considered are:

(i) Total average investment cost, less than or equal to an investment limit.

(ii) The total number of orders per unit time less than a specified amount.

Let for the i^{th} item:

C_i = item cost

λ_i = mean demand per unit time

$\Phi(r_i)$ = probability that the lead time demand is greater than r_i

r_i = reorder point

Q_i = reorder quantity

K_1 = investment limit

K_2 = reorder limit

N = total number of items in inventory

They derive the following mathematical programming problem

$$\underset{Q,r}{\text{Minimize}}\ f \equiv \sum_{i=1}^{N} \frac{B_i(r_i)}{Q_i}$$

subject to

$$g_1 \equiv k_1 - \sum_{i=1}^{N} C_i\ (r_i + \frac{Q_i}{2} - \mu_i) \geq 0,$$

$$g_2 \equiv k_2 - \sum_{i=1}^{N} \frac{\lambda i}{Q_i} \geq 0,$$

$$g_{2+i} \equiv Q_i > 0 \quad \text{for } i = 1, \ldots, N$$

where $\beta_i(r_i) = \frac{1}{2} [\sigma_i^2 + (r_i - \mu_i)^2 \, \Phi \left(\frac{r_i - \mu_i}{\sigma_i}\right) - \frac{\sigma_i}{2} (r_i - \mu_i) \, \phi\left(\frac{r_i - \mu_i}{\sigma_i}\right)$ and

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2/2}{2}} \quad \text{and} \quad \Phi(r) = \int_r^x \phi(x)\,dx$$

TABLE 4

DATA FOR INVENTORY MODEL

N=3	Item 1	Item 2	Item 3
λ_i	1000	1500	2000
C_i	1	10	20
μ_i	100	200	300
σ_i	100	100	200

with $k_1 = 8000$ and $k_2 = 15$

The problem was solved using data given in Table 4 and the results are:

$Q_1 = .5331615\text{E } 03$, $Q_2 = .2455901\text{E } 03$, $Q_3 = .2850370\text{E } 03$

$r_1 = .2527785\text{E } 03$, $r_2 = .2770080\text{E } 03$, $r_3 = .4366119\text{E } 03$

Consumer Expenditure Model

Friedman and Meiselman (1963) estimated consumer expenditure based on the money stock using a simple model

$$C_t = \alpha + \beta M_t + \varepsilon_t \qquad (6.1)$$

where C_t is the consumer expenditure for year t, M_t is its corresponding money stock and ε_t follows a first order autoregressive scheme, i.e., $\varepsilon_t = U_t + \rho\varepsilon_{t-1}$ where ρ is an unknown parameter with $0 < \rho < 1$ and U_t is distributed normally with mean 0 and variance σ_u^2. The logarithmic likelihood function is given by (5.2).)

$$L = \frac{n-1}{2} \log (2r) - \frac{n-1}{2} \log \sigma_u^2 - \frac{1}{2\sigma_u^2} \sum_{t=2}^{n} [(C_t - \rho C_{t-1}) - \alpha(1-\rho)$$

$$- \beta(X_t - \rho X_{t-1})]^2 \qquad (6.2)$$

A heuristic solution to this nonlinear optimization problem was obtained by Kmenta (1971). Table 5 provides the necessary data for estimating the parameters of 6.2. The problem which we solved in its canonical form is:

$$\text{Minimize:} \quad f \equiv (n-1) \log\tau + \frac{1}{\tau} \sum_{t=2}^{n} [(C_t - \rho C_{t-1}) - \alpha(1-\rho) - \beta(X_t - \rho X_{t-1})]^2$$

$$\text{Subject to} \quad g_1 \equiv \rho + .9999 \geq 0$$

$$g_2 \equiv \tau \geq 0$$

$$g_3 \equiv .9999 - \rho \geq 0$$

where $\tau = \sigma^2$

Our algorithm, for varying starting values of r of the penalty function, yielded two local minima. The solution which resulted in the lowest value in objective function was not discovered by Kmenta.

TABLE 5

DATA FOR CONSUMER EXPENDITURE MODEL

Year and Quarter	Consumer Expenditure	Money Stock	Year and Quarter	Consumer Expenditure	Money Stock*
1952 I	214.6	159.3	1954 III	238.7	173.9
II	217.7	161.2	IV	243.2	176.1
III	219.6	162.8			
IV	227.2	164.4	1955 I	249.4	178.0
			II	254.3	179.1
1953 I	230.9	165.9	III	260.9	180.2
II	233.3	167.9	IV	263.3	181.2
III	234.1	168.3			
IV	232.3	169.7	1956 I	265.6	181.6
			II	268.2	182.5
1954 I	233.7	170.5	III	270.4	183.3
II	236.5	171.6	IV	275.6	184.3

The solution which resulted in the lowest value in objective function is

$\sigma_u^2 = 4.372803, \quad \rho = .9999, \quad \alpha = .1896194\text{E } 05, \quad \beta = 1.004103$

The other local minimizer is

$\sigma_u^2 = 4.4638454, \quad \rho = .8240537, \quad \alpha = -.2354883\text{E } 03, \quad \beta = 2.753057$

Even though there is not a substantial difference between the optimal value of the objective function for these two solutions, the economical interpretation of these estimators is no longer identical.

Source: Milton Friedman and David Meiselman (1963)

*In billions of dollars

The solution for this problem was facilitated by selecting a very small value for r_1 (r_1 = 1.0E - 07) .

Criteria for Evaluation of Modified Newton's Methods

In Section 5 three possible strategies of modified Newton methods (S1MNM, S2MNM, S3MNM) were presented. In this section we attempt to study the computational characteristics of each of these methods when the inverse of the Hessian matrix is obtained via the QTU algorithm. Here we determine if any one of these three strategies is superior over the remaining two.

The line search used in S1MNM and S2MNM is the "exact" line search OSSP while in S1MNM we use McCormick's generalization of Armijo's method.

Another pending question which we attempt to answer is which one of the four strategies for selection of descent direction of nonpositive curvature that was discussed in Section 5 is most appropriate to be used with modified Newton methods. In order to measure the computational efficiency of each method we need to specify several criteria for selection of the best algorithm. The most important criteria for a nonlinear programming algorithm is its reliability in obtaining a correct solution. For the purpose of this experiment we selected several well-known test problems with known solutions.

Since all three strategies of modified Newton methods have basically the same procedure when the Hessian matrix is positive definite, one would be interested to examine these methods only for points where the Hessian matrix is nonpositive. Therefore, it is important to count the number of moves along the direction of nonpositive curvature during the solution procedure.

The next important criteria is the number of functions, gradient evaluation when a point with nonpositive Hessian matrix is encountered. Another criteria which we considered is the CPU time spent for each problem. The time which was measured is the actual time used rather than merely the elapsed time from the start of the execution to termination. The former is more desirable since it does not depend on CPU utilization.

A note of caution is in order to warn the reader that these criteria are only used to evaluate a portion of the optimization rather than the overall performance of this code. The following functions were used to carry out the experiments. The alphabetic letters next to each test problem will be used to identify the test problems.

(A) Shell Dual

(B) Gasoline Demand Function

(C) Inventory Model

(D) Consumer Expenditure Model

(E) Wood's Function:

$$\text{minimize:}\quad f(X) \equiv 100\,(X_2 - X_1^2)^2 + (1 - X_1)^2 + 90(X_4 - X_3^2)^2$$

$$+(1-X_3)^2 + 10.1[(X_2-1)^2 + (X_4-1)^2]$$

$$+19.8(X_2-1)(X_4-1)$$

The starting point is (-3,-1,-3,-1)

(F) Rosenbrock's Problem

minimize: $f(X) \equiv (1-X_1)^2 + 100(X_2 - X_1^2)^2$

The starting point is (-1.2, 1.0)

(G) Powell's Function I

minimize: $f(X) \equiv (X_1 + 10X_2)^2 + 5(X_3 - X_4)^2 + (X_2 - 2X_3)^4 + 10(X_1 - X_4)^4$

The starting point is (3.0,-1.0,0.0.1.0)

(H) Powell's Badly Scaled Function

minimize: $f(X) \equiv (10^4 X_1 X_2 - 1)^2 + (e^{-X_1} + e^{-X_2} -1.0001)^2$

The starting point is (0.0, 1.0)

(I) Four Cluster Function

minimize: $f(X) \equiv [(X_1 - X_2^2)\ (X_1 - \sin X_2)]^2 + [(\cos(X_2) - X_1)$

$(X_2 - \cos\ (X_1))]^2$

The starting point is (0.0, 0.0)

(J) Brown's Function with Two Global Minima

minimize: $f(X) \equiv (X_1^2 - X_2 - 1)^2 + ((X_1 - X_2)^2 + (X_2 - 0.5)^2 -1)^2$

The starting point is (0.1,2.0)

(K) Penalty Function I

minimize: $f(X) \equiv 10^{-5} \sum_{i=1}^{4} (X_i - 1)^2 + [\ \sum_{i=1}^{4} X_i^2 - \frac{1}{4}]^2$

The starting point is (1.,2.,3.,4.)

(L) Box's Function

minimize: $f(X) \equiv \sum_{i=1}^{10} \{e^{-x_1 \delta_i} - e^{-x_2 \delta_i} - X_3\ (e^{-\delta_i} - e^{-10\delta_i})\}^2$

The starting point is: (0,20,20)

(M) Gottfried's Function

minimize: $f(X) \equiv (X_1 - .1136(X_1 + 3X_2)\ (1 - X_1))^2 + (X_2 +$

$7.5\ (2X_1 - X_2)\ (1 - X_2))^2$

The starting point is: (0.5,0.5)

(N) Hyperbola-Circle Function

minimize: $f(X) \equiv (X_1X_2 - 1)^2 + (X_1^2 + X_2^2 - 4)^2$

The starting point is: (0.,1.0)

(O) Brown's Badly Scaled Problem

minimize: $f(X) \equiv (X_1 - 10^6)^2 + (X_2 - 2\text{x}10^{-6})^2 + (X_1X_2 - 2)^2$

The starting point is: (1.0, 1.0)

(P) Cubic Function

minimize: $f(X) \equiv X_1^8 + 2\ (\cdot 1X_1^2(X_2 - 1))^2 + (X_2 - 1)^8 + 2(.1X_1^2(X_3 - 1)^2)^2$

$+(X_3 - 1)^8 + 2$

The starting point is: (2.0,-3.0,3.0)

(Q) Rosenbrock Cliff

$$f(X) \equiv [\frac{X_1-3}{100}]^2 - (X_1 - X_2) + e^{20(X_1 - X_2)}$$

The starting point is: (0.0,-1.0)

Experimental Procedure

In Appendix C of Emami (1978) a complete listing of each test problem is given when the solution procedure is based on S1MNM and directions of nonpositive curvatures are evaluated according to DNPC1 strategy.

The convergence criteria which was selected is

$$|\nabla^T P(X,{}^{(i)}r)\ [\nabla^2 P(X,{}^{(i)}r)]^{-1} \nabla P(X,{}^{(i)}r)| < \varepsilon$$

where $P(X,^{(i)} r)$ is the value of the penalty function at point $X^{(i)}$ for parameter r. The values of ε which were used in a majority of the test problems were set equal to approximately 10^{-5}. We selected these relatively large values of ε in order to expedite the termination of the solution. Each computer run was limited by three minutes of CPU time and 3000 printed lines; if it exceeded these limits an interupt would cause the automatic termination of the problem.

The initial values of r for all problems were set equal to one except for the Consumer Expenditure Model for which we select r = 1.E - 07. In order to measure the performance of various strategies under similar conditions, we must ensure that the final value of r is the same for all solution strategies. This is achieved by minimizing the penalty function (P(X,r)) for the initial value of r.

The evaluation of the direction of nonpositive curvature strategies was limited to three strategies. The first strategy, which is based on a computing average direction of nonpositive curvature, could not be evaluated since not many observations were possible among these test problems. It is important to note that the average direction of a nonpositive curvature is itself a direction of nonpositive curvature when the estimate of the positive part of the Hessian matrix is not in full rank and each direction of nonpositive curvature is equal to the last (n-t) rows of Q matrix.

7. Numerical Results

In this section we present the results of numerical experiments which were carried out for various strategies. Each test problem is identified by an alphabetic letter, followed by a two digit number. The first digit after the alphabetic letter identifies the modified Newton's strategy as follows:

1 ≡ S1MNM 2 ≡ S2MNM 3 ≡ S3MNM

The second digit signifies the strategy used for selection of directions of nonpositive curvature where

1 ≡ DNPC1
2 ≡ DNPC2
3 ≡ DNPC3
4 ≡ DNPC4

As an example B23 indicates the solution of gasoline demand when the solution procedure was strategy 2 of modified Newton method and directions of nonpositive curvatures were evaluated by DNPC3 strategy.

The results of our numerical experiments are summarized in Tables 6 through 14 and can be divided into two broad categories. In the first category are the problems where the solution procedure did not generate any points where the Hessian matrix was nonpositive definite. The second category consists of those problems where points were generated having nonpositive definite Hessian matrices. Among the first group, S3MNM performed best since the number of function evaluations were considerably less than the other two strategies. In most instances this coincided with a reduction in execution time. However this method did not perform satisfactorily whenever a point with a nonpositive definite Hessian matrix was encountered. It seems that the algorithm had difficulty in guiding the moves away from the difficult region. This is indicated by the large number of moves along directions of nonpositive curvature (see Table 12, Problems D and E). This result does not fully support Sorenson's result (Sorenson 1977) even though he used a different algorithm for finding a point along $\bar{S}^{(k)} = \alpha S^{(k)} + d^{(k)}$ which minimized $f(X^{(k)} + \alpha\bar{S}^{(k)})$. Before any conclusion can be reached one must evaluate Goldfarb's algorithm (Goldfarb 1977) and compare it with the results of Sorenson and those contained in Tables 6 through 14 of this report. Algorithm S3MNM had difficulties with overflow problems which prohibited it from finding the solution to Problem L.

The performances of strategies one and two were very similar except that the algorithm S1MNM terminated prematurely on Problem I. It is important to note that had we selected the value of ε equal to zero the early termination would not have resulted. However, due to the nature of S2MNM which at each iteration makes one Newtonian move and may or may not move along a direction of nonpositive curvature the chance of premature termination is practically eliminated. On the basis of these observations we recommend S2MNM as a reliable and efficient optimization algorithm.

Our preliminary investigations into direction of nonpositive curvature strategies indicates that there is no substantial difference among them. The test problems which were examined here did not generate many alternative directions of nonpositive curvature. We believe as these directions increase in number the selection of the "best" direction of nonpositive curvature needs special care. However, based on limited number of problems, we recommend DNPC2 over the remaining strategies.

TABLE 6

RESULT OF S1MNM WITH DNPC2

Test Problems and its Solution Strategy	Number of Function Evaluation	Number of Gradient Evaluation	Number of Move Along D.N.P.C.	Final Value of Penalty Function	Total Execution Time	Total Number of Iteration
A12	9704	75	2	.4951548E 02	91.3	25
B12	705	144	2	.2308220E 03	172.6	36
C12	380	15	0	-.8307040E 01	7.2	5
D12	1113	93	2	.2351634E 02	35.4	32
E12	475	69	2	.6718380E-17	9.8	23
F12	264	36	0	.5182552E-12	2.3	12
G12	220	30	0	.2022576E-7	3.1	10
H12	261	36	1	.1365220E-5	3.6	12
I12	26	3	1	.7252676E-1	.5	1
J12	187	24	1	.5901125E-21	2.6	8
K12	74	9	1	.6033433E-04	1.3	3
L12	143	12	2	.6584739E-23	5.3	4
M12	131	15	1	.2875892E-10	1.8	5
N12	157	18	1	.5429854E-22	1.9	6
012	211	21	1	.1355302E-19	2.1	7
P12	127	6	0	.3192631E-60	2.2	3
Q12	68	12	0	.1997866	.9	4

TABLE 7

RESULT OF S1MNM WITH DNPC3

Test Problems and its Solution Strategy	Number of Function Evaluation	Number of Gradient Evaluation	Number of Move Along D.N.C.P.	Final Value of Penalty Function	Total Execution Time	Total Number of Iteration
A13	9704	75	2	.4951548E 02	97.1	25
B13	653	132	2	.3360575E 03	173.0	34
C13	380	15	0	-.830704E 01	8.3	5
D13	1113	93	2	.2351634E 02	35.2	32
E13	475	69	2	.671838E-17	9.9	23
F13	264	36	0	.5182552E-12	2.3	12
G13	220	30	0	.2022576E-7	3.2	10
H13	261	36	1	.1365220E-05	3.6	12
I13	26	3	1	.7252676E-01	.5	1
J13	187	24	1	.5901125E-21	2.2	8
K13	74	9	1	.6033433E-04	1.3	3
L13	143	12	2	.6584739E-23	5.4	4
M13	131	15	1	.2875892E-10	1.8	5
N13	157	18	1	.5429854E-22	1.9	6
O13	211	21	1	.1355302E-19	2.1	7
P13	127	6	0	.3192631E-60	2.2	3
Q13	68	12	0	.1997866	.9	4

TABLE 8

RESULT OF S1MNM WITH DNPC4

Test Problems and its Solution Strategy	Number of Function Evaluation	Number of Gradient Evaluation	Number of Move Along D.N.P.C.	Final Value of Penalty Function	Total Execution Time	Total Number of Iteration
A14	10592	81	3	.4951548E 02	104.5	27
B14	671	136	2	.2849867E 03	173.6	35
C14	380	15	0	-.8307040E 01	8.2	5
D14	1113	93	2	.2351634E 02	34.1	32
E14	475	69	2	.671838E-17	10.3	23
F14	264	36	0	.5182552E-12	2.4	12
G14	220	30	0	.2022576E-07	3.2	10
H14	261	36	1	.136522E-05	3.7	12
I14	26	3	1	.07252676	.6	1
J14	187	24	1	.5901125E-21	2.3	8
K14	74	9	1	.6033433E-4	1.3	3
L14	143	12	2	.6584739E-23	5.4	4
M14	131	15	1	.2875892E-10	1.8	5
N14	157	18	1	.5429854E-22	2.0	6
O14	211	21	1	.1355302E-19	2.1	7
P14	127	6	0	.3192631E-60	2.2	3
Q14	68	12	0	.1997866	.9	4

TABLE 9

RESULT OF S2MNM WITH DNPC2

Test Problems and its Solution Strategy	Number of Function Evaluation	Number of Gradient Evaluation	Number of Move Along D.N.P.C.	Final Value of Penalty Function	Total Execution Time	Total Number of Iteration
A22	11210	83	2	.4951548E 02	109.9	30
B22	608	116	6	.2172173E 03	153.9	37
C22	380	15	0	-.8307040E 01	8.4	5
D22	1164	58	4	.2351634E 02	26.5	24
E22	504	71	2	.3473133E-12	10.9	25
F22	264	36	0	.5182552E-12	2.7	12
G22	220	30	0	.2022576E-07	3.6	10
H22	283	37	1	.1256657E-5	4.2	13
I22	107	10	1	.1444885E-15	1.7	4
J22	144	16	1	.3004309E-18	1.8	6
K22	96	10	1	.6023866E-04	1.7	4
L22	215	16	4	.1843696E-06	8.0	8
M22	120	8	2	.4065190E-05	1.3	4
N22	237	20	2	.5327739E-22	2.5	9
O22	190	16	1	.3049319E-17	1.9	6
P22	127	6	0	.3192631E-60	2.2	3
Q22	68	12	0	.1997866	.9	4

TABLE 10

RESULT OF S2MNM WITH DNPC3

Test Problems and its Solution Strategy	Number of Function Evaluation	Number of Gradient Evaluation	Number of Move Along D.N.P.C.	Final Value of Penalty Function	Total Execution Time	Total Number of Iteration
A23	11210	83	2	.4951548E 02	108.8	30
B23	710	132	6	.2172173E 03	173.7	39
C23	380	15	0	-.8307040E 01	8.1	5
D23	1164	58	4	.2351634E 02	26.1	24
E23	504	71	2	.3473133E-12	10.9	25
F23	264	36	0	.5182552E-12	2.6	12
G23	220	30	0	.2022576E-7	3.5	10
H23	283	37	1	.1256657E-5	4.1	13
I23	107	10	1	.1444885E-15	1.7	4
J23	144	16	1	.3004309E-18	1.8	6
K23	96	10	1	.6023866E-04	1.6	4
L23	215	16	4	.1843696E-06	7.6	8
M23	120	8	2	.4065190E-05	1.3	4
N23	237	20	2	.5327739E-22	2.4	9
O23	190	16	1	.3049319E-17	1.9	6
P23	127	6	0	.3192631E-60	2.2	3
Q23	68	12	0	.1997866	.9	4

TABLE 11

RESULT OF S2MNM WITH DNPC4

Test Problems and its Solution Strategy	Number of Function Evaluation	Number of Gradient Evaluation	Number of Move Along D.N.P.C.	Final Value of Penalty Function	Total Execution Time	Total Number of Iteration
A24	11291	81	3	.4951548E 02	100.5	30
B24	724	140	6	.2172173E 03	175.0	41
C24	380	15	0	-.8307040E 01	7.7	5
D24	1164	58	4	.2351634E 02	26.6	22
E24	504	71	2	.3473133E-12	11.0	25
F24	264	36	0	.5182552E-12	2.7	12
G24	220	30	0	.2022576E-7	3.6	10
H24	283	37	1	.1256657E-5	4.0	13
I24	107	10	1	.1444885E-15	1.7	4
J24	144	16	1	.3004309E-18	1.8	6
K24	96	10	1	.6023866E-4	1.7	4
L24	215	16	4	.1843696E-6	7.8	8
M24	120	8	2	.4065190E-5	1.3	4
N24	237	20	2	.5327739E-22	2.5	9
024	190	16	1	.3049319E-17	1.9	6
P24	127	6	0	.3192631E-60	2.2	3
Q24	68	12	0	.1997860	.9	4

TABLE 12

RESULT OF S3MNM WITH DNPC2

Test Problems and its Solution Strategy	Number of Function Evaluation	Number of Gradient Evaluation	Number of Move Along D.N.P.C.	Final Value of Penalty Function	Total Execution Time	Total Number of Iteration
A32	1370	163	1	.4951548E 02	162.9	54
B32	59	172	43	.2902303E 03	176.4	43
C32	78	21	0	-.8307039E 01	7.9	7
D32	1013	357	27	.2351634E 02	105.6	110
E32	53	149	14	.1308768E-15	16.2	45
F32	29	63	0	.4684773E-14	3.1	21
G32	15	39	0	.1122226E-14	2.7	13
H32	31	76	1	.2027482E-05	5.8	25
I32	25	32	2	.3151770E-08	3.1	10
J32	13	32	2	.3643502E-14	2.5	10
K32	16	33	0	.2381209E-04	3.1	11
L32				FAILED		
M32	19	28	1	.3206540E-08	2.5	9
N32	12	20	2	.7560333E-24	1.7	6
O32	50	76	1	.3604638E-22	5.7	25
P32	24	66	0	.1090929E-06	6.8	22
Q32	29	81	0	.1997866	4.1	27

TABLE 13
RESULT OF S3MNM WITH DNPC3

Test Problems and its Solution Strategy	Number of Function Evaluation	Number of Gradient Evaluation	Number of Move Along D.N.P.C.	Final Value of Penalty Function	Total Execution Time	Total Number of Iteration
A33	1370	163	1	.4951548E 02	169.9	54
B33	56	164	41	.3556660E 03	175.8	41
C33	78	21	0	-.8307039E 01	8.2	6
D33	738	322	19	.2351634E 02	96.4	101
E33	53	149	14	.1308768E-15	16.2	45
F33	29	63	0	.4684773E-15	3.2	21
G33	15	39	0	.1122226E-6	2.9	13
H33	31	76	1	.2027482E-5	5.9	25
I33	58	40	4	.3280797E-8	3.9	12
J33	13	32	2	.3643502E-14	2.5	10
K33	16	33	0	.2381209E-04	3.1	11
L33				FAILED		
M33	19	28	1	.3206540E-8	2.4	9
N33	12	20	2	.7560333E-24	1.7	6
O33	50	76	1	.3604638E-22	5.7	25
P33	24	66	0	.1090929E-6	6.8	22
Q33	29	81	0	.1997866	4.1	27

TABLE 14

RESULT OF S3MNM WITH DNPC4

Test Problems and its Solution Strategy	Number of Function Evaluation	Number of Gradient Evaluation	Number of Move Along D.N.P.C.	Final Value of Penalty Function	Total Execution Time	Total Number of Iteration
A34	1147	133	1	.4951548E 02	143.7	44
B34	50	140	35	.8477251E 03	155.4	35
C34	78	21	0	-.8307039E 01	9.1	7
D34	863	336	33	.2351634E 02	99.5	101
E34	53	149	14	.1308768E-15	16.1	45
F34	29	63	0	.4684773E-14	3.2	21
G34	15	39	0	.1122226E-6	2.9	13
H34	31	76	1	.1539316E-4	5.9	25
I34	58	40	4	.3280797E-8	3.9	12
J34	13	32	2	.3643502E-14	2.5	10
K34	16	33	0	.2381209E-4	3.1	11
L34				FAILED		
M34	19	28	1	.3206540E-08	2.5	9
N34	12	20	2	.7560333E-24	1.7	6
034	50	76	1	.3604638E-22	5.7	25
P34	24	66	0	.1090929E-06	6.8	22
Q34	29	81	0	.1997866	4.1	27

8. Conclusions

Several papers have recently investigated the problem of altering Newton's method when the Hessian matrix is not positive definite. The first attempt in this area was in Chapter 8 of Fiacco and McCormick 1968. Recently Fletcher and Freeman 1975, Gill and Murray 1974, Goldfarb 1977, and Sorenson 1977 have proposed ideas for this situation. Direct comparison of their results with ours is difficult because in most cases the problems used were different and because it is hard to generate examples on which comparisons can be made. Usually (as seen in the Tables) only one or two cases requiring modification occur.

The results of this study were inconclusive in many respects. Certainly the obtaining of estimates of the positive part and directions of nonpositive curvature operating on the factored Hessian are now established as an alternative to the decomposition of the Hessian matrix in its symmetric n by n form. Which of the basic strategies is best to use (S1MNM, S2MNM, or S3MNM) and which direction of nonpositive curvature is best has not been answered. It does appear that the Second Order Armijo Step Size procedure does not move enough in the direction of the nonpositive curvature and requires many more moves to minimize a nonconvex function. It does best in the positive definite case requiring fewer function evaluations and computer time. This is misleading however since an inefficient optimal step-size procedure was used for (5.1).

An idea which came out of this research and may prove fruitful is a modification of the basic algorithm given in Section 5. When $K_j < 0$ occurs, instead of ignoring that rank one matrix, a better strategy might be to absorb as much of it as possible without making the resulting matrix have a negative eigenvalue. The formula is simply

$$P_{j+1} = P_j + a_j(-a_j^T P_j^+ a_j)^{-1} a_j^T P_j \ .$$

This modification would have the advantage of simplicity and reduce the number of kinds of directions of nonpositive curvature considered. This will be tested in the future.

REFERENCES

[1] BENNETT, J. M. and D. R. GREEN (1966). Updating the inverse of the triangular factors of a modified matrix. Technical Report No. 42. Basser Computing Department, University of Sydney.

[2] EMAMI, G. (1978). Evaluating strategies for Newton's method using a numerically stable generalized inverse algorithm. D.Sc. Dissertation. School of Engineering and Applied Science, The George Washington University.

[3] FIACCO, A. V. and G. P. McCORMICK (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Wiley, New York.

[4] FLETCHER, R. F. and T. L. FREEMAN (1975). A modified Newton method for minimization. Report No. 7. Department of Mathematics, University of Dundee.

[5] FRIEDMAN, M. and D. MEISELMAN (1963). The relative stability of monetary velocity and the investment multiplier in the United States, 1897-1958. *Commission on Money and Credit, Stabilization Policies*. Prentice-Hall, New Jersey.

[6] GILL, P. E. and W. MURRAY (1974). Newton-type method for unconstrained and linearly constrained optimization. *Mathematical Programming*. 7 311-350.

[7] GOLDFARB, DONALD (1977). Curvilinear path steplength algorithms for minimization which use directions of negative curvature. Report CCNY-CS-77-101. School of Engineering, City College of the City University of New York.

[8] KMENTA, JAN (1971). *Elements of Econometrics*. Macmillan Press, New York.

[9] McCORMICK, G. P. (1974). A mini-manual for the use of the SUMT computer program and factorable programming language. Technical Report SOL-74-15. System Optimization Laboratory, Department of Operations Research, Stanford University.

[10] SCHRADY, D. A. and U. C. CHOE (1971). Models for multi-item continuous review inventory policies subject to constraints. *Naval Res. Logist. Quart.* **18** 451-463.

[11] SORENSON, D. C. (1977). Updating the symmetric indefinite factorization with application in a modified Newton's method. ANL-77-49. Argonne National Laboratory.

THE GEORGE WASHINGTON UNIVERSITY
Program in Logistics
Distribution List for Technical Papers

- The George Washington University
  - Office of Sponsored Research
  - Library
  - Vice President H. F. Bright
  - Dean Harold Liebowitz
  - Mr. J. Frank Doubleday
- ONR
  - Chief of Naval Research
    - (Codes 200, 430D, 1021P)
    - Resident Representative
- OPNAV
  - OP-40
  - DCNO, Logistics
  - Navy Dept Library
  - OP-911
  - OP-964
- Naval Aviation Integrated Log Support
- NAVCOSSACT
- Naval Cmd Sys Sup Activity Tech Library
- Naval Electronics Lab Library
- Naval Facilities Eng Cmd Tech Library
- Naval Ordnance Station
  - Louisville, Ky.
  - Indian Head, Md.
- Naval Ordnance Sys Cmd Library
- Naval Research Branch Office
  - Boston
  - Chicago
  - New York
  - Pasadena
  - San Francisco
- Naval Research Lab
  - Tech Info Div
  - Library, Code 2029 (ONRL)
- Naval Ship Engng Center
  - Philadelphia, Pa.
  - Hyattsville, Md.
- Naval Ship Res & Dev Center
- Naval Sea Systems Command
  - Tech Library
  - Code 073
- Naval Supply Systems Command
  - Library
  - Capt W. T. Nash
- Naval War College Library
  - Newport
- BUPERS Tech Library
- FMSO
- Integrated Sea Lift Study
- USN Ammo Depot Earle
- USN Postgrad School Monterey
  - Library
  - Dr. Jack R. Borsting
  - Prof C. R. Jones
- US Marine Corps
  - Commandant
  - Deputy Chief of Staff, R&D
- Marine Corps School Quantico
  - Landing Force Dev Ctr
  - Logistics Officer
- Armed Forces Industrial College
- Armed Forces Staff College
- Army War College Library
  - Carlisle Barracks
- Army Cmd & Gen Staff College
- US Army HQ
  - LTC George L. Slyman
  - Army Trans Mat Command
- Army Logistics Mgmt Center
  - Fort Lee
- Commanding Officer, USALDSRA
  - New Cumberland Army Depot
- US Army Inventory Res Ofc
  - Philadelphia
- HQ, US Air Force
  - AFADS-3
- Griffiss Air Force Base
  - Reliability Analysis Center
- Maxwell Air Force Base Library
- Wright-Patterson Air Force Base
  - HQ, AF Log Command
  - Research Sch Log
- Defense Documentation Center
- National Academy of Science
  - Maritime Transportation Res Board Library
- National Bureau of Standards
  - Dr E. W. Cannon
  - Dr Joan Rosenblatt
- National Science Foundation
- National Security Agency
- WSEG
- British Navy Staff
- Logistics, OR Analysis Establishment
  - National Defense Hdqtrs, Ottawa
- American Power Jet Co
  - George Chernowitz
- ARCON Corp
- General Dynamics, Pomona
- General Research Corp
  - Dr Hugh Cole
  - Library
- Planning Research Corp
  - Los Angeles
- Rand Corporation
  - Library
- Carnegie-Mellon University
  - Dean H. A. Simon
  - Prof G. Thompson
- Case Western Reserve University
  - Prof B. V. Dean
  - Prof John R. Isbell
  - Prof M. Mesarovic
  - Prof S. Zacks
- Cornell University
  - Prof R. E. Bechhofer
  - Prof R. W. Conway
  - Prof J. Kiefer
  - Prof Andrew Schultz, Jr.
- Cowles Foundation for Research
  - Library
  - Prof Herbert Scarf
  - Prof Martin Shubik
- Florida State University
  - Prof R. A. Bradley
- Harvard University
  - Prof K. J. Arrow
  - Prof W. G. Cochran
  - Prof Arthur Schleifer, Jr.
- New York University
  - Prof O. Morgenstern
- Princeton University
  - Prof A. W. Tucker
  - Prof J. W. Tukey
  - Prof Geoffrey S. Watson

Purdue University
Prof S. S. Gupta
Prof H. Rubin
Prof Andrew Whinston

Stanford
Prof T. W. Anderson
Prof G. B. Dantzig
Prof F. S. Hillier
Prof D. L. Iglehart
Prof Samuel Karlin
Prof G. J. Lieberman
Prof Herbert Solomon
Prof A. F. Veinott, Jr.

University of California, Berkeley
Prof R. E. Barlow
Prof D. Gale
Prof Rosedith Sitgreaves
Prof L. M. Tichvinsky

University of California, Los Angeles
Prof J. R. Jackson
Prof Jacob Marschak
Prof R. R. O'Neill
Numerical Analysis Res Librarian

University of North Carolina
Prof W. L. Smith
Prof M. R. Leadbetter

University of Pennsylvania
Prof Russell Ackoff
Prof Thomas L. Saaty

University of Texas
Prof A. Charnes

Yale University
Prof F. J. Anscombe
Prof I. R. Savage
Prof M. J. Sobel
Dept of Admin Sciences

Prof Z. W. Birnbaum
University of Washington

Prof B. H. Bissinger
The Pennsylvania State University

Prof Seth Bonder
University of Michigan

Prof G. E. P. Box
University of Wisconsin

Dr. Jerome Bracken
Institute for Defense Analyses

Prof H. Chernoff
MIT

Prof Arthur Cohen
Rutgers – The State University

Mr Wallace M. Cohen
US General Accounting Office

Prof C. Derman
Columbia University

Prof Paul S. Dwyer
Mackinaw City, Michigan

Prof Saul I. Gass
University of Maryland

Dr Donald P. Gaver
Carmel, California

Dr Murray A. Geisler
Logistics Mgmt Institute

Prof J. F. Hannan
Michigan State University

Prof H. O. Hartley
Texas A & M Foundation

Mr Gerald F. Hein
NASA, Lewis Research Center

Prof W. M. Hirsch
Courant Institute

Dr Alan J. Hoffman
IBM, Yorktown Heights

Dr Rudolf Husser
University of Bern, Switzerland

Prof J. H. K. Kao
Polytech Institute of New York

Prof W. Kruskal
University of Chicago

Prof C. E. Lemke
Rensselaer Polytech Institute

Prof Loynes
University of Sheffield, England

Prof Steven Nahmias
University of Pittsburgh

Prof D. B. Owen
Southern Methodist University

Prof E. Parzen
State University New York, Buffalo

Prof H. O. Posten
University of Connecticut

Prof R. Remage, Jr.
University of Delaware

Dr Fred Rigby
Texas Tech College

Mr David Rosenblatt
Washington, D. C.

Prof M. Rosenblatt
University of California, San Diego

Prof Alan J. Rowe
University of Southern California

Prof A. H. Rubenstein
Northwestern University

Dr M. E. Salveson
West Los Angeles

Prof Edward A. Silver
University of Waterloo, Canada

Prof R. M. Thrall
Rice University

Dr S. Vajda
University of Sussex, England

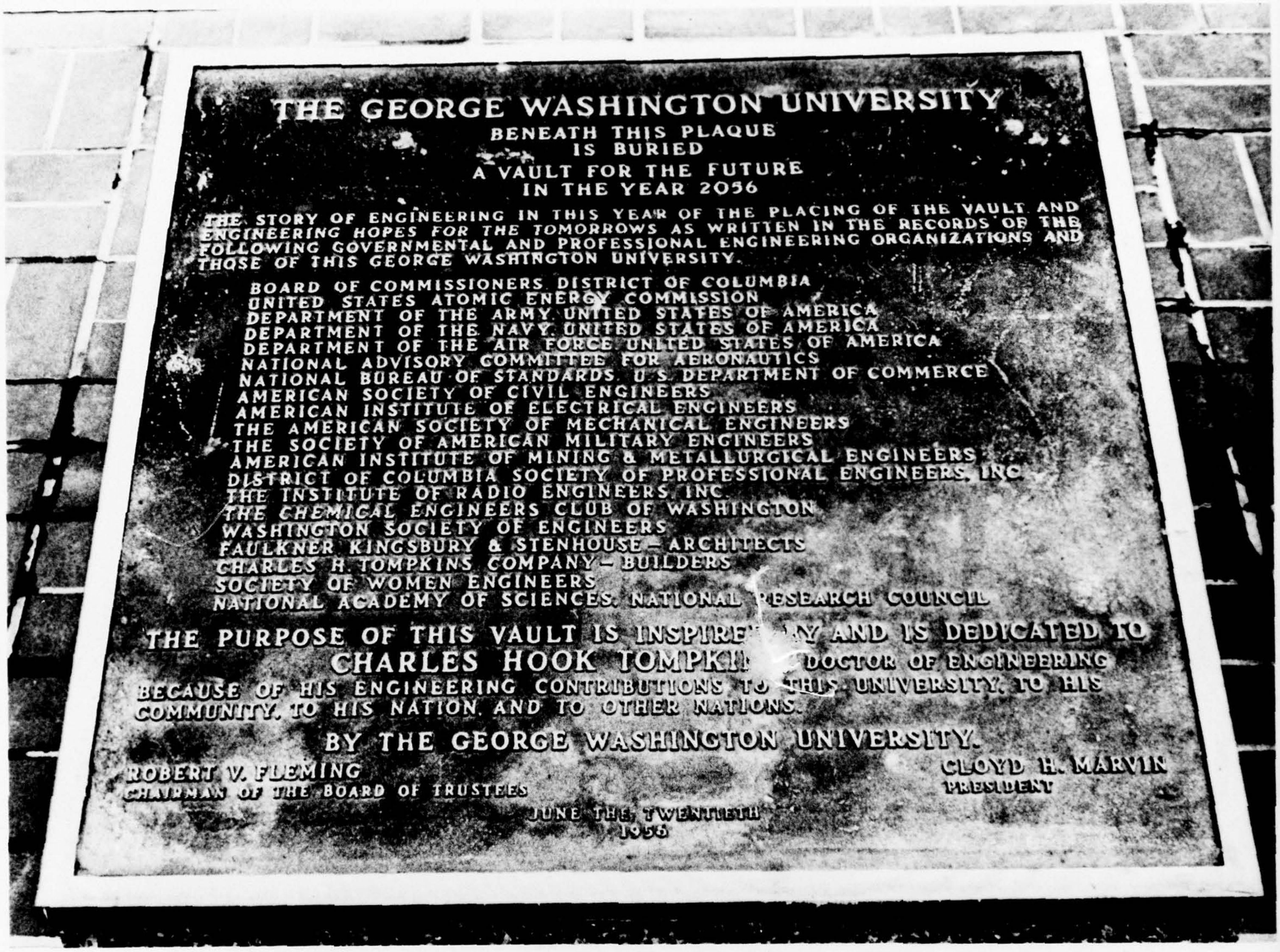
Prof T. M. Whitin
Wesleyan University

Prof Jacob Wolfowitz
University of Illinois

Mr Marshall K. Wood
National Planning Association

Prof Max A. Woodbury
Duke University

May 1976



THE GEORGE WASHINGTON UNIVERSITY

BENEATH THIS PLAQUE
IS BURIED
A VAULT FOR THE FUTURE
IN THE YEAR 2056

THE STORY OF ENGINEERING IN THIS YEAR OF THE PLACING OF THE VAULT AND ENGINEERING HOPES FOR THE TOMORROWS AS WRITTEN IN THE RECORDS OF THE FOLLOWING GOVERNMENTAL AND PROFESSIONAL ENGINEERING ORGANIZATIONS AND THOSE OF THIS GEORGE WASHINGTON UNIVERSITY.

BOARD OF COMMISSIONERS DISTRICT OF COLUMBIA
UNITED STATES ATOMIC ENERGY COMMISSION
DEPARTMENT OF THE ARMY UNITED STATES OF AMERICA
DEPARTMENT OF THE NAVY UNITED STATES OF AMERICA
DEPARTMENT OF THE AIR FORCE UNITED STATES OF AMERICA
NATIONAL ADVISORY COMMITTEE FOR AERONAUTICS
NATIONAL BUREAU OF STANDARDS U.S. DEPARTMENT OF COMMERCE
AMERICAN SOCIETY OF CIVIL ENGINEERS
AMERICAN INSTITUTE OF ELECTRICAL ENGINEERS
THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS
THE SOCIETY OF AMERICAN MILITARY ENGINEERS
AMERICAN INSTITUTE OF MINING & METALLURGICAL ENGINEERS
DISTRICT OF COLUMBIA SOCIETY OF PROFESSIONAL ENGINEERS, INC
THE INSTITUTE OF RADIO ENGINEERS, INC.
THE CHEMICAL ENGINEERS CLUB OF WASHINGTON
WASHINGTON SOCIETY OF ENGINEERS
FAULKNER KINGSBURY & STENHOUSE – ARCHITECTS
CHARLES H. TOMPKINS COMPANY – BUILDERS
SOCIETY OF WOMEN ENGINEERS
NATIONAL ACADEMY OF SCIENCES, NATIONAL RESEARCH COUNCIL

THE PURPOSE OF THIS VAULT IS INSPIRED BY AND IS DEDICATED TO
CHARLES HOOK TOMPKINS DOCTOR OF ENGINEERING
BECAUSE OF HIS ENGINEERING CONTRIBUTIONS TO THIS UNIVERSITY, TO HIS COMMUNITY, TO HIS NATION, AND TO OTHER NATIONS.

BY THE GEORGE WASHINGTON UNIVERSITY.

ROBERT V. FLEMING
CHAIRMAN OF THE BOARD OF TRUSTEES

CLOYD H. MARVIN
PRESIDENT

JUNE THE TWENTIETH
1956

To cope with the expanding technology, our society must be assured of a continuing supply of rigorously trained and educated engineers. The School of Engineering and Applied Science is completely committed to this objective.